



## インテル® Xeon® プロセッサ向け 並列プログラミング・ツール・ガイド

ホワイトペーパー

**Q: アプリケーションのパフォーマンスを向上させるために最適なプログラミング・モデルとツールをどのように選択するべきでしょうか?**

### はじめに

マイクロプロセッサのパフォーマンス・ゲインの主な要因がクロックスピードからマルチコア、より幅の広いベクトル、増加するベクトル命令セットなどの機能に移行するのに伴い、開発者がアプリケーションを最適化してハードウェア固有の機能を活用することの重要性が増しています。しかし、そのような機能をフルに活用するための専門ツールがあることをご存じでしょうか?

このホワイトペーパーでは、ハードウェア機能を紹介し、ツールの選択肢と各ツールの詳細を説明し、開発者が自身のアプリケーションに最適なツールを特定できるように質問と回答を記述します。あなたが、C、C++、または Fortran で開発を行っており、いくらかのコードの変更を行う用意があるのであれば、ぜひこのホワイトペーパーをお読みになり、インテル® ソフトウェア開発製品によって、企業がアプリケーションの大幅なパフォーマンス向上をどのように達成しているかを参考にしてください。

### 並列化を実装する理由

並列化を実装する理由はさまざまですが、並列化を行う動機と求めている目標を理解することが重要です。これまでの経験からすると、企業は次のような理由で並列化を行います。

- アプリケーションのパフォーマンスに関して顧客から不満の声がある
- タイム・クリティカルなソリューションを導き出すアプリケーションである (例: 次の日の天気予報に1週間はかけられない)
- 競合製品が並列化を実装している
- アプリケーションに新しい機能を追加する (例: AES 暗号化をプロセッサが行うことで全体のパフォーマンスへの影響を最小限に抑えつつアプリケーションのセキュリティーを改善する)

### インテル® Xeon® プロセッサのパフォーマンス指向の機能

インテル® Xeon® プロセッサには複数の固有の機能があり、アプリケーションのパフォーマンスを大幅に向上させることができます。しかし、これらの機能を活用してその恩恵を受けるには、プログラミングの専門知識と業界標準またはインテルのツールが必要です。

プロセッサの機能	アプリケーションが得られる利点
インテル® スマート・キャッシュ	キャッシュミスを減らすことにより、高速なプロセッサ・メモリーを利用
インテル® ハイパースレディング・テクノロジー	マルチスレッド・コードを複数の仮想コアで実行することにより、パフォーマンスを向上
インテルのマルチコア・テクノロジー	マルチスレッド・コードを複数の物理コアで実行することにより、パフォーマンスを向上
インテル® SSE (インテル® AVX も含む)	ベクトル化されたデータを各コアで並列に実行することにより、パフォーマンスを向上
インテル® AES-NI	ハードウェアによるセキュリティー機能を使用した保護
パフォーマンス・モニタリング・ユニット (PMU)	PMU によりキャッシュミスなどのパフォーマンス問題を発見

## 利用可能なツール

インテルでは、並列化を実装するための幅広いプログラミング・ツールとモデルを用意しています。

### プログラミング・ツール

解析ツールは、エラーやセキュリティーの脆弱性を探ることによりソースコードの問題の箇所を特定し、判断のための手がかりを提供します。ライブラリーは定義済み関数群を提供します。この関数群は、コードに簡単に追加できるため、インテル® プロセッサーでより効率良くパフォーマンス指向の機能を利用できます。コンパイラーは、最適化機能とスレッド化を提供します。クラスターツールは、メッセージ・パッシング・インターフェイス (MPI) を使用するクラスター向けの開発において、並列アプリケーションのパフォーマンス解析と最適化を支援します。スイート製品は、複数またはすべてのツールが 1 つのパッケージにまとまっているので、簡単にインストールできます。

カテゴリー	インテル製品	サポート言語
解析ツール	インテル® Parallel Advisor	C、C++
	インテル® VTune™ Amplifier XE	C/C++、Fortran、C#
	インテル® Inspector XE	C/C++、Fortran
ライブラリー	インテル® マス・カーネル・ライブラリー (インテル® MKL)	C/C++、Fortran
	インテル® インテグレートッド・パフォーマンス・プリミティブ (インテル® IPP)	C/C++。C#、Java* 向けの使用サンプル
	インテル® スレディング・ビルディング・ブロック (インテル® TBB)	C++ テンプレート・ライブラリー
コンパイラー	インテル® Composer XE	C/C++、Fortran
クラスターツール	インテル® MPI ライブラリー	C/C++、Fortran
	インテル® Trace Analyzer/Collector	C/C++、Fortran
スイート製品	インテル® Parallel Studio XE	C/C++、Fortran
	インテル® Cluster Studio XE	C/C++、Fortran

### プログラミング・モデル

インテルは、マルチコア・プロセッサーとメニーコア・プロセッサーの両方で使用できる並列プログラミング・モデルの共通セットを提供しています。各モデルを利用して、開発者はアプリケーションに並列処理を実装したり、並列性を向上させることができます。プログラミング・モデルは、高レベルに抽象化され簡単に使用できるように設計されたもの (インテル® Cilk™ Plus、インテル® TBB など) から、高度に最適化されて広範囲にスレッド化され導入が容易なライブラリー (インテル® IPP、インテル® MKL)、コンピューティング環境に応じてスケールし、既存の標準規格を実装するツール (MPI、OpenMP\*、Co-Array Fortran、OpenCL\*) まで幅広く揃っています。次の表に、提供されるモデルをまとめています。

インテル® Cilk™ Plus	インテル® スレディング・ビルディング・ブロック	ドメイン固有のライブラリー	既存の標準規格
並列化が容易な C/C++ 言語拡張	並列化するための一般的な C++ テンプレート・ライブラリー	インテル® インテグレートッド・パフォーマンス・プリミティブ インテル® マス・カーネル・ライブラリー	メッセージ・パッシング・インターフェイス (MPI) OpenMP* Co-Array Fortran OpenCL*

プログラミング・モデル: 幅広いツールを開発者に提供  
アプリケーションのパフォーマンスを最適化すべきかどうかを  
判断

## アプリケーションのパフォーマンスを最適化すべきかどうかの判断

必要な機能と可能性を引き出すための最適なツールの組み合わせは、アプリケーションと環境に依存します。インテルのコンサルタント・エンジニアは、並列化を行うべきかどうかを判断し、最適なツールを提案するために、通常、次のような質問をします。

質問	対応
あなたはソースコードにアクセスできるソフトウェア開発者ですか?	「はい」の場合は、続行
パフォーマンスを向上するために労力を惜しみませんか?	「はい」の場合は、続行
コードのパフォーマンスを解析したことがありますか?	<ul style="list-style-type: none"> <li>コードをプロファイリングして、時間を費やしている箇所、時間を無駄にしている箇所、待機が長すぎる箇所を特定することから開始します。インテル® VTune™ Amplifier XE は、インテル® Xeon® プロセッサのオンチップ・パフォーマンス・モニタリング・ユニット (PMU) を使用してハードウェアのイベントサンプリングを行います。例: オンボードキャッシュ (インテル® スマート・キャッシュ・テクノロジー) の非効率的な使用を特定する。</li> <li>インテル® Parallel Advisor: 段階的に並列化候補を特定し、実装を試していくことができます。</li> <li>インテル® Inspector XE: デッドロックや競合状態などの発見が困難な問題を含む並列処理問題を引き起こすメモリ/スレッドエラーを発見します。コードが正しくスレッド化され、仮想コア (インテル® ハイパースレディング・テクノロジー) と物理コア (インテル® のマルチコア・テクノロジー) の両方にわたって実行されることを保証します。</li> </ul>
関数の hotspot を見つけましたか?	<p>遅いライブラリー関数を、(低価格でロイヤリティー無料の) 高度にチューニングされ並列化されたバージョンに置き換えます。特定の関数 (BLAS、FFT、行列乗算など) が hotspot の場合は、インテルが提供するライブラリーをぜひご利用ください。インテルのライブラリーには、<u>何千もの</u>高度に最適化された定義済み関数群が含まれており、コードにただ追加するだけです。また、再コンパイルすることなく、将来のインテル® プロセッサに<b>対応</b>します。</p> <p>インテル® マス・カーネル・ライブラリー (インテル® MKL): 工学、科学、金融系アプリケーション向け関数群を提供します。</p> <p>(例: インテル® AVX による最適化を利用することで、インテル® MKL はインテル® Xeon® プロセッサ上で SMP LINPACK のパフォーマンスを最大 90% まで向上します。)</p> <p>インテル® インテグレートッド・パフォーマンス・プリミティブ (インテル® IPP): 画像、ビデオ、通信と信号処理、データ処理向けの関数群を提供します。(例: インテル® Xeon® プロセッサに搭載されたインテル® AES-NI 命令セットを使用することで、インテル® IPP は OpenSSL のパフォーマンスを最大 145% まで向上します。)</p>
モジュールの hotspot を見つけましたか?	1 つのモジュール内で hotspot を見つけた場合は、その部分だけをインテル® コンパイラーでコンパイルし、残りは既存のコンパイラーを引き続き使用することができます。また、自動ベクトル化や他の自動機能も試してみてください。
パフォーマンスをさらに向上したいですか?	次ページに示すプログラミング・モデルを含むツールキットを検討してみてください。

注: インテルが提供するツールは、特に浮動小数点演算を多用するアプリケーションに最適です。

## アプリケーションと環境に合致したプログラミング・モデルの特定

次に互換性について見てみましょう。

質問	対応
使用するプログラミング言語は何ですか?	<p>C - インテル® Cilk™ Plus、インテル® IPP、インテル® MKL、(可能性として) インテルの OpenCL* SDK</p> <p>C++ - インテル® Cilk™ Plus、インテル® TBB、インテル® IPP、インテル® MKL</p> <p>Fortran - Co-Array Fortran、インテル® MPI、インテル® MKL、インテル® IPP (クロス言語の呼び出し経由)</p> <p>C# - インテル® MKL およびインテル® IPP の一部の API のサポート</p> <p>スクリプト言語 (Python*、PERL*) - インテル® MKL およびインテル® IPP を使用した場合の限定されたオプション</p>
必要な OS サポートは?	<p>Windows* - 包括的なサポート</p> <p>Linux* - 包括的なサポート</p> <p>Mac OS* X - インテル® C++ Composer XE およびインテル® Fortran Composer XE (インテル® Cilk™ Plus、インテル® TBB、インテル® IPP、インテル® MKL、コンパイラー、OpenMP*、Co-Array Fortran) を使用してサポート</p>

注:すべてのツールは、インテル® アーキテクチャーおよび互換システムでサポートされます。

プログラミング・モデルの選択:

質問	対応
新しいプログラミング・モデルをどの程度習得したいですか? コードの変更をどの程度行ってもかまわないですか?	<p>あまりしたくない場合 - インテル® IPP およびインテル® MKL ライブラリーを使用</p> <p>ある程度してもかまわない場合 - ライブラリーに加えて、インテル® Cilk™ Plus、インテルの OpenMP*、Co-Array Fortran を検討</p> <p>非常に前向きな場合 - 最大限の柔軟性とパフォーマンスを備えるインテル® TBB、インテル® MPI を検討</p>
共有メモリー、クラスター、ベクトル・プログラミングを使用していますか?	<p>クラスターを使用するプログラミング - 分散メモリーを使うアプリケーションには MPI を使用。MPI を使用するノードには OpenMP*、インテル® Cilk™ Plus、インテル® TBB を使用。</p> <p>共有メモリーを使用するプログラミング - OpenMP*、インテル® Cilk™ Plus、インテル® TBB を使用。</p> <p>ベクトルを使用するプログラミング (SSE/AVX など) - インテル® Cilk™ Plus、Fortran90 アレイ・ノーテーションを使用。</p>
CPU + GPU ハイブリッド・プログラミングは重要ですか?	「はい」の場合は、OpenCL* の使用を検討

## プログラミング・ツールとモデル

**インテル® Parallel Studio XE (スイート製品):** 業界最高レベルの C/C++ コンパイラーと Fortran コンパイラー、パフォーマンス・ライブラリーと並列ライブラリー、エラーチェッカー、パフォーマンス・アナライザーなどのテクノロジーを組み合わせたツール・スイートです。ハイパフォーマンス・コンピューティングやエンタープライズ・アプリケーションで必要とされるパフォーマンス、そしてコードの品質やセキュリティ、信頼性を高めるのに役立ちます。同時に、このスイートさえあれば、ハイパフォーマンスを実現するために必要なすべてのツールを調達する手間もなく、共通のツールセットを使ってマルチコアから将来のメニーコアへの移行をスムーズに行えます。詳細は、[software.intel.com/en-us/articles/intel-parallel-studio-xe/](http://software.intel.com/en-us/articles/intel-parallel-studio-xe/) を参照してください。

**インテル® Cluster Studio XE (スイート製品):** インテル® Parallel Studio XE にクラスターツールが追加されています。この製品は、IA-32 およびインテル® 64 アーキテクチャー・ベースの Windows\*/Linux\* プラットフォームで共有、分散、ハイブリッド・メモリーを使用する C++/Fortran アプリケーションの開発と解析において、パフォーマンス、信頼性、スケーラビリティを最大限に引き出す初のツール・スイートです。詳細は、[software.intel.com/en-us/articles/intel-cluster-studio-xe/](http://software.intel.com/en-us/articles/intel-cluster-studio-xe/) を参照してください。

**インテル® Parallel Advisor (解析):** C/C++ プログラムへ並列処理を実装したり、並列化を向上するように開発者を支援します。インテル® Parallel Advisor を使用すると、開発者は、最も時間を費やしているシリアルコード領域を特定し、アノテーションを挿入して並列化の実装を試し、提案された変更の適応性をチェックし、並列化した場合にアプリケーションが正しく動作するかどうかを確認できます。ツールは、現在は Microsoft\* Windows\* 向けのみです。詳細は、[software.intel.com/en-us/articles/intel-parallel-advisor/](https://software.intel.com/en-us/articles/intel-parallel-advisor/) を参照してください。

**インテル® Composer XE (コンパイラーとライブラリー):** インテル® C++ コンパイラー、インテル® Fortran コンパイラー、インテル® マス・カーネル・ライブラリー (インテル® MKL)、インテル® インテグレートッド・パフォーマンス・プリミティブ (インテル® IPP)、インテル® スレディング・ビルディング・ブロック (インテル® TBB) が含まれるスイート製品です。C/C++ および Fortran 開発者は、最新のインテル® アーキテクチャー・プロセッサ上で動作するハイパフォーマンスなエンタープライズ・アプリケーションを開発、保守できます。インテル® Composer XE でただ再コンパイルするだけで、20 % 以上ものパフォーマンス向上を達成することが可能です。パフォーマンスの向上は、メモリーの最適化、自動並列化、ベクトル化によってもたらされます。詳細は、[software.intel.com/en-us/articles/intel-composer-xe/](https://software.intel.com/en-us/articles/intel-composer-xe/) を参照してください。

**インテル® VTune Amplifier XE (解析ツール):** シリアル動作と並列動作を理解し、パフォーマンスとスケーラビリティの向上を目指す C/C++、Fortran、C# 開発者向けの強力なスレッド/パフォーマンス・プロファイリング・ツールです。Windows\* および Linux\* アプリケーションのパフォーマンス・アナライザーとして、インテル® VTune™ Amplifier XE は迅速かつ精度の高い意思決定を促すスケーリング情報を素早く提供し、推測による作業を回避します。コアが十分に有効活用され、新しいプロセッサ機能が完全にサポートされるように、最適なパフォーマンスを導くきめ細かなチューニングを行います。ソフトウェア機能には多くの新しい事前定義済みパフォーマンス・プロファイリングが用意されており、マイクロアーキテクチャーの詳細を知らなくても有益なプロファイリング情報を簡単に収集できます。プロファイリング後、タイムライン、フィルタリング、フレーム解析などの解析機能によってデータが変換され、実用的な情報が示されます。詳細は、[www.intel.com/software/products/vtune](http://www.intel.com/software/products/vtune) を参照してください。

**インテル® Inspector XE (解析ツール):** このツールは、メモリー/スレッドエラーを検出することにより、アプリケーションの信頼性を高めます。Windows\* および Linux\* 向けソフトウェアを作成する C、C++、C#、Fortran 開発者向けに設計されています。メモリー解析ツールは、メモリーリークやメモリー破壊を開発段階の早期に発見できます。スレッド解析ツールとデバッガーは、問題となる相互作用を持つスレッドやデータ競合およびデッドロックを見つけます。エラーを引き起こすシナリオが実行されない場合でも、間欠的なエラーや非決定性のエラーを発見できます。このツールを開発段階で使用することにより、セキュリティ問題を早めに検出し、解決できます。このよう

な問題は、アプリケーションがいったん導入されてから解決するのは、非常にコストがかかります。詳細は、[software.intel.com/en-us/articles/intel-inspector-xe/](https://software.intel.com/en-us/articles/intel-inspector-xe/) を参照してください。

**インテル® MPI ライブラリー (クラスター・ライブラリー):** ハイパフォーマンスな MPI-2 規格を複数のファブリックに実装することにより、インテル® アーキテクチャー・ベースのクラスターでアプリケーションのパフォーマンスを向上させます。ソフトウェアや動作環境を大きく変更することなく、インターコネクトを変更したり、新しいインターコネクトにアップグレードできます。ハイパフォーマンスなインテル® MPI ライブラリーを使用することで、実行時にユーザーによって選択される複数のクラスター・ファブリック・インターコネクトで実行可能なアプリケーションを開発することができます。インテル® MPI ライブラリーで開発された製品用に無償のランタイム環境も提供しています。エンタープライズ向け、デバイス向け、部門向け、ワークグループ向けのハイパフォーマンス・コンピューティングにそのクラス最高のパフォーマンスを提供します。詳細は、[software.intel.com/en-us/articles/intel-mpi-library/](https://software.intel.com/en-us/articles/intel-mpi-library/) を参照してください。

**インテル® Trace Analyzer および Trace Collector (クラスター解析):** MPI アプリケーションの正当性と動作を理解するのに役立つ強力なツールで、アプリケーションのイベントベース・トレース機能を提供する、オーバーヘッドの極めて少ないトレース・ライブラリーが含まれます。収集したトレースデータを解析して、パフォーマンスの hotspot やボトルネックを発見することができます。この製品は、完全にスレッドセーフで、MPI の使用にかかわらず、C/C++、FORTRAN、およびマルチスレッド・プロセスに用いることができます。インテル® Trace Collector はバイナリー・インストールメンテーションとフェイルセーフ・モードをサポートします。また、MPI プログラミング・エラーとシステムエラーをチェックします。インテル® Trace Analyzer は、インテル® Trace Collector によって収集されたアプリケーションのアクティビティをグラフィカルに表示して分析できる便利な方法を提供します。任意のレベルで詳細を表示し、パフォーマンスの hotspot とボトルネックを素早く識別して、原因を分析します。1 つにバンドルされたインテル® Trace Analyzer/Collector は、最適化された解析機能と視覚化機能を提供します。両ツールにより、複雑なプロファイリング・データを高速にグラフィック・レンダリングすることができ、また何百ものプロセスを簡単にトレースできます。ツールは、Linux\* 版および Windows\* 版があります。詳細は、[software.intel.com/en-us/articles/intel-trace-analyzer/](https://software.intel.com/en-us/articles/intel-trace-analyzer/) を参照してください。

## プログラミング・モデル

**ライブラリー (インテル® マス・カーネル・ライブラリーおよびインテル® インテグレートッド・パフォーマンス・プリミティブ):** ライブラリーは、プログラミングを開始する前に考慮すべき重要な抽象並列プログラミング手法を提供します。BLAS、ビデオ、オーディオ・エンコーダー/デコーダー、FFT、ソルバーおよびソーターを含むアルゴリズムを備えたライブラリーの実装について検討することは重要です。インテルの提供するライブラリーは、SSE および AVX 命令セット、マルチコア/メニーコア・プロセッサを活用するべく高度にチューニングされた多くのアルゴリズムの高度な実装を提供します。ライブラリーのルーチンへの1つの呼び出しで、単一ソースコードはこれらの利点が得られます。インテル® MKL は Fortran の標準インターフェイスを持ち、新しい業界標準の LAPACK の C インターフェイスもサポートします。インテルのライブラリーに対するハイパフォーマンスの飽くなき追求により、並列プログラミングの第1候補としてライブラリーが簡単な選択肢となっています。詳細は、[www.intel.com/software/products/mkl/](http://www.intel.com/software/products/mkl/) および [www.intel.com/software/products/ipp/](http://www.intel.com/software/products/ipp/) を参照してください。

### インテル® MKL を使用した Fortran の 行列乗算

```
call
DGEMM(transa,transb,m,n,k,alpha,a,lda,b,
ldb,beta,c,ldc)
```

**インテル® Cilk™ Plus:** 迅速かつ容易で信頼性のある C および C++ 言語拡張で、タスク、データ、ベクトル並列化を効果的に使用するための拡張されたタスク処理と配列表記 (アレイ・ノーテーション) を提供します。Cilk は、最初に米マサチューセッツ工科大学 (MIT) でレイサーソン教授らによって研究開発され、その後彼が設立した Cilk Arts 社で C++ に実装されました。インテル® Cilk™ Plus は、プロセッサおよびコプロセッサでタスク、データ、ベクトル並列化を求める C/C++ 開発者に奨励されます。ベクトル並列化の利点を活用するプロジェクトでは、インテル® Cilk™ Plus はインテル® TBB の前に検討してください。インテル® Cilk™ Plus とインテル® TBB の機能は、混在して使用できます。そのため、インテル® TBB を使用しているプログラムで、インテル® Cilk™ Plus のデータ/ベクトル並列機能を使うことができます。詳細は、<http://cilkplus.org> を参照してください。

### インテル® Cilk™ Plus を使用した C の 並列関数の呼び出し

```
cilk_for (int i=0; i<n; ++i) {
    Foo(a[i]);
}
```

**インテル® スレディング・ビルディング・ブロック (インテル® TBB):** C++ 並列プログラミング向けの最もポピュラーな抽象化手法を提供します。2006 年にインテルよりリリースされ、2007 年にオープン・ソース・プロジェクトになりました。業界でも数多く採用され、OpenMP\* に勝る人気を誇っています。例えば、インテル® TBB を採用している企業として、Adobe、Autodesk、Dreamworks Animation などがあります。インテル® TBB は数多くのオペレーティング・システムとプロセッサに移植されており、マルチコアやメニーコアにおける並列処理を求める C++ 開発者に最適です。詳細は、<http://threadingbuildingblocks.org> を参照してください。

### インテル® TBB を使用した C++ の 並列関数の呼び出し

```
parallel_for (0, n,
    [=](int i) { Foo(a[i]); }
);
```

**OpenMP\*:** 1996 年に OpenMP\* 規格は、コンパイラーによる並列ハードウェアの活用を支援する方法として提唱されました。そして、10 年以上が経過した今では、主要な C、C++、Fortran コンパイラーは OpenMP\* をサポートしています。OpenMP\* は特に Fortran プログラムと、C で記述された科学プログラムのニーズに適しています。インテルは、OpenMP\* グループのメンバーであり、OpenMP\* 実装とサポートツールにおける先駆者です。OpenMP\* はマルチコア/メニーコア・プログラミングに使用できます。詳細は、<http://openmp.org> を参照してください。

### OpenMP\* を使用した C のベクトル要素の合計

```
#pragma omp parallel for reduction(+: s)
for (int i = 0; i < n; i++) {
    s += x[i];
}
```

**メッセージ・パッシング・インターフェイス (MPI):** クラスタを使用する開発者にとって、メッセージ・パッシング・インターフェイス (MPI) は最も一般的なプログラミング手法です。クラスタでは、メッセージを渡すことによりプロセッサが接続されます(ただし、メモリーは共有であるとは限らない)。クラスタのノードにメニーコア・プロセッサが搭載されているかどうかにかかわらず、通信には MPI が使用されます。広く使用されているインテル® MPI ライブラリーは、ほぼすべてのインターコネクトに対してハイパフォーマンスとサポートを提供します。マルチコア/メニーコア・プロセッサ・ベースのシステムに対応し、現在の MPI プログラミングと同じ慣れ親しんだ方法でマルチコア/メニーコア・プロセッサ上でランクを作成します。詳細は、<http://intel.com/go/mpl> を参照してください。

#### C の MPI コード (クラスタ用)

```
for (d=1; d<ntasks; d++) {
    rows = (d <= extra) ? avrow+1 : avrow;
    printf(" sending %d rows to task %d\n",
    rows, dest);
    MPI_Send(&offset, 1, MPI_INT, d, mtype,
    MPI_COMM_WORLD);
    MPI_Send(&rows, 1, MPI_INT, d, mtype,
    MPI_COMM_WORLD);
    MPI_Send(&a[offset][0], rows*NCA,
    MPI_DOUBLE, d, mtype, MPI_COMM_WORLD);
    MPI_Send(&b, NCA*NCB, MPI_DOUBLE, d,
    mtype, MPI_COMM_WORLD);
    offset = offset + rows;
}
```

明するソリューションで最もリファクタリングが必要なものでしょう。特にハードウェアの深い知識に基づいたリファクタリングが必要です。リファクタリング作業の結果は、マルチコア/メニーコアのパフォーマンスで顕著かもしれません。結果のパフォーマンスはそのようなリファクタリングなしでも可能である場合があります。OpenCL\* の目標は、リファクタリングに取り組む際に生産性の高い方法で行うことです。OpenCL\* 以外のソリューションでは、ハードウェアの深い知識に基づいたリファクタリングの必要性を避ける手法を提供します。詳細は、<http://intel.com/go/opencl> を参照してください。

#### OpenCL\* を使用した C の要素ごとの乗算

```
kernel void
dotprod(    global const float *a,
           global const float *b,
           global float *c) {
    int myid = get_global_id(0);
    c[myid] = a[myid] * b[myid];
}
```

**Co-Array Fortran:** Fortran 2008 規格では、堅固で効率的な並列プログラミング言語の追加の手法として “Co-Array” が含まれました。インテル® Fortran コンパイラーは、共有および分散の両方のメモリーシステムで Fortran 2008 規格の Co-Array を使用する並列プログラミングをサポートします。Co-Array Fortran は SPMD (Single Program Multiple Data) プログラミング・モデルを使用します。詳細は、<http://intel.com/software/products> を参照してください。

#### Co-Array を使用した Fortran

```
REAL SUM[*]
CALL SYNC_ALL( WAIT=1 )
DO IMG= 2,NUM_IMAGES()
    IF (IMG==THIS_IMAGE()) THEN
        SUM = SUM + SUM[IMG-1]
    ENDIF
CALL SYNC_ALL( WAIT=IMG )
ENDDO
```

**OpenCL\*:** 最初にアップル社により提案され、その後、業界標準化団体に移されました。インテルはこの団体に参加しています。OpenCL\* は、ハードウェアに近いインターフェイスを備え、幅広い業界の関心と取り組みによって生み出された重要な抽象化と十分な制御を提供します。OpenCL\* は、このホワイトペーパーで説

## まとめ

パフォーマンスの問題は棚上げしないでください。インテル® プロセッサが新しいパフォーマンス指向の機能を引き続き搭載するのに伴い、ソースコードもベクトル化/スレッド化が必要になります。このホワイトペーパーでは、解析ツールがどのようにパフォーマンス向上へつながる部分を特定し、遅い関数がどのように高度に最適化されたインテルの関数に置き換えられ、モジュール内の hotspot がどのようにインテル® コンパイラーで再コンパイルされ、さまざまなプログラミング・モデルがどのように開発者をサポートするかを説明しました。

インテルでは、25 年以上にわたり、インテル® プロセッサの機能を活用するべく開発者にプログラミング・ツールを提供し続けています。我々は、アプリケーションには多くのニーズがあることを理解しており、開発者に幅広いツールの選択肢と、さまざまなプログラミング問題を解決するためのオプションを提供します。インテルのツールを使用したユーザーからは、10%、2 倍、ときには 20 倍ものパフォーマンス・ゲインを達成したという声が寄せられています。ツールは、Microsoft\* Visual Studio\*、Eclipse\*、さらに GCC\* コンパイラーにも対応するように設計されています。

インテル® メニー・インテグレートッド・コア (インテル® MIC) アーキテクチャー・ベースの製品の登場に備えて、その処理能力をフルに活用できるようにソースコードの準備を整えておくことがこれから重要になってきます。幸いにも、マルチコアに適用される標準化されたハイレベルな x86 プログラミング・モデルは、メニーコアにも適用されます。インテル® Xeon® プロセッサ向けにいったん最適化すれば、開発者は、その既存コードとプログラミングの専門知識をインテル® MIC アーキテクチャーにもまた活用できます。ぜひ、[software.intel.com/parallel/](https://software.intel.com/parallel/) の情報もご覧ください。また、[software.intel.com/en-us/forums/threading-on-intel-parallel-architectures/](https://software.intel.com/en-us/forums/threading-on-intel-parallel-architectures/) のフォーラムでは質問を投稿できます。

## 評価版

製品の評価版をダウンロードしてお試しく下さい。初めてのユーザーには、次の製品から開始することをお勧めします。

1. スイート製品: [インテル® Parallel Studio XE](#) または [インテル® Cluster Studio XE](#)
2. [インテル® VTune™ Amplifier XE](#) パフォーマンス・アナライザー
3. [インテル® Parallel Advisor](#) Windows\* 版

## ラーニングツール

インテルのラーニングラボでは、チュートリアルやホワイトペーパーなどが掲載されています。

## 推奨するドキュメント

- [マルチスレッド・アプリケーション開発のためのガイド](#)
- Web セミナー - 「[The Key to Scaling Applications for Multicore](#) (マルチコア向けにアプリケーションをスケールリングするための鍵)」
- [インテル® VTune™ Amplifier XE - 概要ビデオと詳細な手順の入門チュートリアル](#)
- 記事: [インテル® Parallel Composer を利用して並列コードを開発する](#) (詳細なコードサンプル付き)
- [インテル® TBB 書籍](#) (Amazon.co.jp)

製品情報および購入情報は、インテル® ソフトウェア開発製品 Web サイトを参照してください。

<http://www.intel.co.jp/software/products/>

## 購入方法: 言語別のスイート

アプリケーションをビルド、検証、チューニングする複数のツールが組み合わされた次のスイートがご利用になれます。ライセンスは、シングルユーザー・ライセンス、フローティング・ライセンス、アカデミック・ライセンスが用意されています。

スイート>>	インテル® Parallel Studio XE	インテル® C++ Studio XE	インテル® Fortran Studio XE	インテル® Composer XE	インテル® C++ Composer XE	インテル® Fortran Composer XE
インテル® C/C++ コンパイラー	●	●		●	●	
インテル® Fortran コンパイラー	●		●	●		●
インテル® インテグレートド・パフォーマンス・プリミティブ <sup>3</sup>	●	●		●	●	
インテル® マス・カーネル・ライブラリー <sup>3</sup>	●	●	●	●	●	●
インテル® Cilk™ Plus	●	●		●	●	
インテル® スレディング・ビルディング・ブロック	●	●		●	●	
インテル® Inspector XE	●	●	●			
インテル® VTune™ Amplifier XE	●	●	●			
スタティック・セキュリティー解析	●	●	●			
インテル® MPI ライブラリー						
インテル® Trace Analyzer/Collector						
Rogue Wave IMSL* ライブラリー <sup>2</sup>						●
オペレーティング・システム <sup>1</sup>	W、L	W、L	W、L	W、L	W、L、M	W、L、M

注: (1) オペレーティング・システム: W=Windows\*, L=Linux\*, M=Mac OS\* X. (2) インテル® Visual Fortran Composer XE Windows\* 版 IMSL\* 同梱で利用可能 (3) Mac OS\* X 版は個別に提供されません。インテル® C++/Fortran Composer XE Mac OS\* X 版に含まれています。

### 著者について

Ashish Gupta はインテル コーポレーションのプロダクト・マネージャーで、さまざまなソフトウェア・ツールのマーケティングを担当しています。

### 著作権と商標について

本資料に掲載されている情報は、インテル製品の概要説明を目的としたものです。本資料は、明示されているか否かにかかわらず、また禁反言によるとよらずにかかわらず、いかなる知的財産権のライセンスを許諾するものではありません。製品に付属の売買契約書『Intel's Terms and Conditions of Sale』に規定されている場合を除き、インテルはいかなる責任を負うものではなく、またインテル製品の販売や使用に関する明示または黙示の保証 (特定目的への適合性、商品適格性、あらゆる特許権、著作権、その他知的財産権の非侵害性への保証を含む) に関してもいかなる責任も負いません。

### 最適化に関する注意事項

改訂 #20110804

インテル® コンパイラーは、互換マイクロプロセッサ向けには、インテル製マイクロプロセッサ向けと同等レベルの最適化が行われない可能性があります。これには、インテル® ストリーミング SIMD 拡張命令 2 (インテル® SSE2)、インテル® ストリーミング SIMD 拡張命令 3 (インテル® SSE3)、ストリーミング SIMD 拡張命令 3 補足命令 (SSSE3) 命令セットに関連する最適化およびその他の最適化が含まれます。インテルでは、インテル製ではないマイクロプロセッサに対して、最適化の提供、機能、効果を保証していません。本製品のマイクロプロセッサ固有の最適化は、インテル製マイクロプロセッサでの使用を目的としています。インテル® マイクロアーキテクチャーに非固有の特定の最適化は、インテル製マイクロプロセッサ向けに予約されています。この注意事項の適用対象である特定の命令セットの詳細は、該当する製品のユーザー・リファレンス・ガイドを参照してください。

